HAW HAMBURG

Bachelorarbeit

Malte Martin Koch

Kooperatives Machine Learning mit DIAL: Untersuchungen zur Agentenkommunikation

Fakultät Technik und Informatik
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Malte Martin Koch

Kooperatives Machine Learning mit DIAL: Untersuchungen zur Agentenkommunikation

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Michael Neitzke
Zweitgutachter: Prof. Dr. Ing. Birgit Wendholt

Eingereicht am: 26. März 2021

Malte Martin Koch

Thema der Arbeit

Kooperatives Machine Learning mit DIAL: Untersuchungen zur Agentenkommunikation

Stichworte

Kooperativ, Maschinelles Lernen, DIAL, Kommunikation

Kurzzusammenfassung

Der Inhalt dieser Bachelorarbeit ist die Analyse von Differential Inter-Agent Learning. Dafür wird ein Schalterrätsel verwendet. Die Agenten kommunizieren miteinander um das Rätsel erfolgreich zu lösen. Dabei verwenden sie unterschiedlichen Nachrichtenlängen. Insbesondere wird hierbei auf das Lernverhalten von Agenten eingegangen, wenn sie zusammen mit Agenten trainieren, die vorher bereits trainiert wurden.

Malte Martin Koch

Title of Thesis

Cooperative Machine Learning with DIAL: Research on agent communication

Keywords

Cooperative, Machine Learning, DIAL, Communication

Abstract

The content of this thesis is the analysis of differential inter-agent learning. For this purpose a switch-riddle environment is used. The agents communicate to solve the riddle. Two different message lenghts are used for the communication. The behaviour of the agents, when trained with pre-trained agents is tested in particular.

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungen

AmV Agent mit Vorwissen.

DIAL Differentiable Inter-Agent Learning.

DQN Deep Q-Network.

DRQN Deep Recurrent Q-Networks.

DRU discretise/regularise unit.

GRU Gated Recurrent Unit.

MDP Markov Decision Process.

MLP Multi-Layer-Perzeptron.

POMDP Partially Observable Markov Decision Process.

RIAL Reinforced Inter-Agent Learning.

RMSProp Root Mean Square Propagation.

RNN Recurrent Neural Network.

TD-Learning Temporal Difference Learning.

1 Einleitung

1.1 Motivation

Die Bedeutung von maschinellem Lernen hat in der Informatik in den letzten Jahren stark zugenommen. Ein Teilaspekt hiervon ist das Reinforcement learning. Die Firma Deepmind hat 2013 erstmals ihren Deep-Q-learning Algorithmus vorgestellt und damit zu einer großen Neuerung in diesem Bereich geführt, indem sie neuronale Netze mit Q-learning kombiniert hat. Auf diese Weise haben sie einer KI beigebracht, verschiedene Atarispiele nur anhand der Bildinformationen zu lernen.[5] Dies war zuvor Aufgrund des großen Zustandsraumes nicht in diesem Ausmaß möglich.
Multi-Agent-Kommunikation ist hierbei der Schlüssel, um kooperative Problemstellungen zu lösen. Dabei ist unter anderem die Länge der ausgetauschten Nachrichten von Bedeutung. 2016 wurden von Foerster et al. die Algorithmen Reinforced Inter-Agent Learning (RIAL) und Differentiable Inter-Agent Learning (DIAL) vorgestellt, mit denen die Agenten lernen sollen zu kommunizieren. Hierbei hat sich DIAL erfolgreicher als RIAL herausgestellt.[2]
Welche Auswirkungen verschiedene Signalgrößen und das Auswechseln von Agenten auf das Lernverhalten und die Kommunikation mit DIAL hat, wird in dieser Arbeit untersucht.

1.2 Zielsetzung

Ziel der Arbeit ist es, herauszufinden, welchen Einfluss die Länge der übermittelten Nachrichten (Signalgröße), sowie das Auswechseln von Agenten mit Vorwissen (AmV), d. h. Agenten die bereits gelernt haben ein Environment zu lösen, durch Agenten ohne Vorwissen, also Agenten die noch nichts gelernt haben, auf das Lernverhalten mit DIAL hat. Dabei werden werden folgende Hypothesen überprüft:

Hypothese 1 *Wenn ein Teil der Agenten ein Kommunikationsprotokoll gelernt hat, lernen die neuen Agenten dieses Protokoll schneller, als wenn alle Agenten kein Vorwissen haben.*

Hypothese 1a *Dieser Lerneffekt ist bei größeren Signalgrößen stärker ausgeprägt.*

Hypothese 2 *Wenn Agenten mit größeren Signalgrößen kommunizieren, brauchen sie mehr Trainingsepisoden um eine Aufgabe zu lösen, als bei einer Kommunikation mit kleineren Signalgrößen.*

Hypothese 3 *Wenn AmV am Training beteiligt sind, lernen die Agenten ohne Vorwissen deren Kommunikationsprotokoll.*

Hypothese 4 *Wenn Agenten mit unterschiedlichen Kommunikationsprotokollen am Training beteiligt sind, verschlechtert dies die Lerngeschwindigkeit.*

1.3 Aufbau der Arbeit

Die für diese Arbeit essentiellen Bereiche des maschinellen Lernens und die verwendeten Techniken werden in Kapitel 2 beschrieben.
In Kapitel 3 wird erläutert, wie die in Abschnitt 1.2 aufgestellten Hypothesen überprüft werden. Insbesondere wird in Abschnitt 3.2 der interne Aufbau der Agenten beschrieben.
Danach werden in Kapitel 4 die jeweiligen Ergebnisse der in Kapitel 3 beschriebenen Versuche dargelegt und analysiert.
Eine Zusammenfassung aller Ergebnisse findet in Kapitel 5 statt.

2 Grundlagen

2.1 Markov Decision Process

Bei einem Markov Decision Process (MDP) handelt es sich um die Formalisierung eines Entscheidungsproblems. Bei diesem haben Aktionen sowohl Einfluss auf eine direkte Belohnung, als auch auf zukünftige Zustände und Belohnungen.[7, S. 47] Der Lernende wird bei einem MDP als Agent bezeichnet. Er wählt bei jedem Zeitschritt eine Aktion (action) und teilt diese der Umgebung (Environment) mit. Diese verarbeitet die Aktion und teilt dem Agenten beim nächsten Zeitschritt seinen daraus resultierenden nächsten Zustand (state) und seine Belohnung (reward) für die gewählte Aktion mit (siehe Abbildung 2.1). Ziel eines Agenten ist es, seine erhaltene Belohnung zu maximieren. Eine Belohnung ist eine reelle Zahl.[7, S. 47 f.] Die Interaktion zwischen Agent und

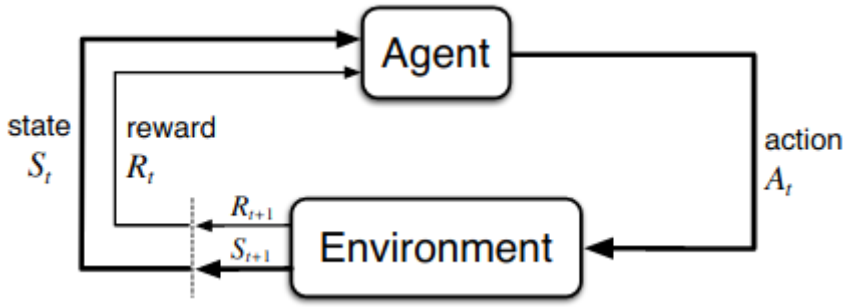

Abbildung 2.1: Die Agenten-Environment Interaktion bei einem Markov Decision Process [7, S. 48]

Environment kann sowohl episodenhaft (z. B. eine Partie Schach), als auch durchgängig sein (z. B. ein Roboter, der eine langfristige Aufgabe erledigt). Zu Beginn befindet sich ein Agent in einem Startzustand, den er vom Environment mitgeteilt bekommt. Dieser Zustand unterscheidet sich nicht von den anderen Zuständen. Ein Zustand ist eine Repräsentation des Environments. Er darf nur von dem Zustand und der Aktion

des letzten Zeitschritts abhängen. Er muss außerdem unabhängig von vorherigen Zuständen und Aktionen betrachtet werden können. Es darf also kein Gedächtnis bei den Agenten vorausgesetzt werden. Ein Zustand muss dementsprechend alle Informationen enthalten, die wichtig für zukünftige Zustände sind. Wenn ein Zustand diese Bedingungen erfüllt, hat er die Markoveigenschaft (Markov property). Diese ist für einen MDP zwingend erforderlich.[7, S. 49]

2.2 Deep Q-Learning

Q-Learning ist ein Temporal Difference Learning (TD-Learning) Algorithmus. Beim TD-Learning wird die Interaktion zwischen Environment und Agenten als MDP beschrieben. TD-Learning Algorithmen passen ihre Strategie während einer laufenden Episode an. Sie müssen also nicht auf das Ende einer Episode warten, um Anpassungen vorzunehmen. [7, S. 119] Beim Q-Learning werden die Aktionen eines Agenten mithilfe einer „action-value function“ bewertet. Die Bewertung erfolgt durch die Formel

$$Q^{neu}(s_t, u_t) \leftarrow Q(s_t, u_t) + \alpha * (r_t + \gamma * max(Q(s_{t+1}, u_t)) - Q(s_t, u_t)),$$

wobei s der momentane Zustand, u die gewählte Aktion, r die erhaltene Belohnung, α die Lernrate und γ der Diskontierungsfaktor des Agenten ist. Bei t handelt es sich um den Zeitschritt. Die „action-value function“ $Q(s_t, u_t)$ errechnet sich also aus dem bisherigen Wert von Q, der Belohnung des Agenten und dem nächsten Wert von Q. Definierend für Q-Learning ist dabei, dass nicht der Q-Wert für die tatsächlich im nächsten Schritt gewählte Aktion verwendet wird. Stattdessen wird der maximal mögliche Q-Wert für den nächsten Zustand gewählt. [7, S. 131]
Q-Learning geht für das Lernen einer Strategie davon aus, dass ein Agent die bisher beste Aktion wählt, unabhängig davon, ob der Agent dies tatsächlich tut. Mit der Lernrate α kann beeinflusst werden, wie stark neue Erkenntnisse sich auf bisher Gelerntes auswirken. Bei einer Lernrate von 1 wird beispielsweise nur das neue Wissen verwendet, d. h. alle früheren Schritte beeinflussen den neuen Wert nicht. Der Diskontierungsfaktor γ beeinflusst, wie stark Belohnungen, die in der Zukunft erwartet werden, sich auf das Handeln des Agenten auswirken. Ein Diskontierungsfaktor von 0 bedeutet, dass der Agent nur anstrebt, die direkt erhaltene Belohnung r zu maximieren, da $max(Q(s_{t+1}, u_t))$ auf Null gesetzt wird.
Beim Deep Q-Learning wird der momentane Q-Wert durch ein neuronales Netz, ei-

nem Deep Q-Network (DQN) berechnet, statt z. B. aus einer Tabelle abgelesen zu werden. Dadurch ist es möglich Q-Learning auf Probleme mit riesigen Zustandsräumen anzuwenden, da für jeden Zustand bereits ein Q-Wert existiert, auf den bisherige Erfahrungen Einfluss genommen haben. Die Parameter des neuronalen Netzes werden durch θ repräsentiert. Aus $Q(s, u)$ wird somit $Q(s, u; \theta)$. Der mit der oben genannten Formel neu berechnete Q-Wert wird für die Kostenfunktion des neuronalen Netzes verwendet. Diese lautet $(y^{DQN} - Q(s, u; \theta))^2$. [6, S. 1]
Um eine bessere Stabilität während des Trainings zu erhalten, wird für den Wert $max(Q(s_{t+1}, u_t)$ der Formel ein anderes DQN verwendet. Dieses Netz hat den exakt gleichen Aufbau wie das ursprüngliche DQN (source network) und startet mit den gleichen Parametern. Es wird als „target network“ bezeichnet. Die Parameter des target network werden nicht trainiert. Stattdessen werden sie nach einer vorher festgelegten Anzahl an Trainingsschritten durch die des source network ersetzt. Aus $max(Q(s_{t+1}, u_t)$ wird $max(Q(s_{t+1}, u_t, \theta'))$. θ' repräsentiert hierbei die Parameter des target network. [6, S. 1]

2.3 Deep Recurrent Q-Networks

DQNs setzen voraus, dass es sich bei einer Aufgabe um einen MDP handelt. Wenn vergangene Zustände den aktuellen Zustand oder die aktuelle Belohnung beeinflussen, handelt es sich stattdessen um einen Partially Observable Markov Decision Process (POMDP). Der Agent erhält keinen Zustand s, sondern eine von s abhängige Beobachtung o. Ein POMDP ist eine allgemeinere Version des MDP. Ein MDP lässt sich in ein POMDP umformen, in dem man die Beobachtungen mit den Zuständen gleich setzt. Eine Möglichkeit teilweise beobachtbare Umgebungen mit DQNs zu kombinieren, sind Deep Recurrent Q-Networks (DRQNs). Um das Problem der teilweisen Beobachtbarkeit anzugehen, verleiht man dem DQN mithilfe eines Recurrent Neural Network (RNN) ein Gedächtnis.[3, S. 1 f.] Statt sich an $Q(s_t, u_t)$ anzunähern, nähert sich der Agent an $Q(o_t, u_t)$ an. Der interne Zustand des RNN lässt sich durch den zusätzlichen Eingabeparameter h_{t-1} repräsentieren, sodass man $Q(o_t, h_{t-1}, u_t)$ erhält. Wenn mehrere Agenten an einer Aufgabe beteiligt sind, die als MDP gilt, muss jeder Agent den gleichen Zustand s der Umgebung bekommen. Bei einem POMDP bekommt stattdessen jeder Agent seine individuelle Beobachtung o des Zustands s der Umgebung. [2, S. 3]

2.4 Differentiable Inter-Agent Learning

Differentiable Inter-Agent Learning ist ein Algorithmus, der es Agenten ermöglichen soll, zu lernen miteinander zu kommunizieren. Die Anzahl an Agenten, die miteinander kommunizieren, ist nicht vorgegeben. Es ist für Agenten sowohl möglich, ihre Nachrichten an alle anderen Agenten zu senden, als auch Nachrichten nur an spezifische Agenten weiterzuleiten. Wenn ein Agent seine Nachrichten an verschiedene andere Agenten schickt, kann er jedem Agenten eine individuelle Nachricht schicken. Nachrichten sind aus Zahlen zusammengesetzt. Eine Nachricht mit einer Länge (Signalgröße) von vier Stellen besteht also aus vier Zahlen. Die Signalgröße einer Nachricht wird vom Algorithmus nicht vorgegeben. Jedoch sollte für alle Agenten immer eine einheitliche Signalgröße verwendet werden.[1]
Bei DIAL handelt es sich um einen POMDP. Für die Agenten werden DRQNs verwendet. In der folgenden Erklärung von DIAL wird davon ausgegangen, dass ein Agent für alle anderen Agenten die gleiche Nachricht erzeugt. Wie bei einem MDP wählt jeder Agent a zu jedem Zeitpunkt t eine Aktion u_t. Bei DIAL wird jedoch zusätzlich noch eine Nachricht m_t gewählt (siehe Abbildung 2.2). Die Aktion wird, wie bei einem MDP üblich, an das Environment geleitet. Die Nachricht wird stattdessen an die sogenannte discretise/regularise unit (DRU) übergeben. Die DRU wandelt die Nachricht um und gibt sie an die anderen Agenten weiter. Sie ist beim Zeitschritt $t + 1$ ein Teil der Eingabeparameter der anderen Agenten. Die Agenten sind dementsprechend über den Kommunikationskanal miteinander verbunden. Da sich die Nachricht nicht von anderen Parametern des neuronalen Netzes unterscheidet, kann der Empfänger einen Gradienten für sie berechnen. Dafür wird zuerst der Gradient für die Aktion wie bei einem DQN berechnet (siehe Abschnitt 2.2). Durch Backpropagation erhält der Agent nun auch einen Gradienten für die Eingabeparameter und somit auch der empfangenen

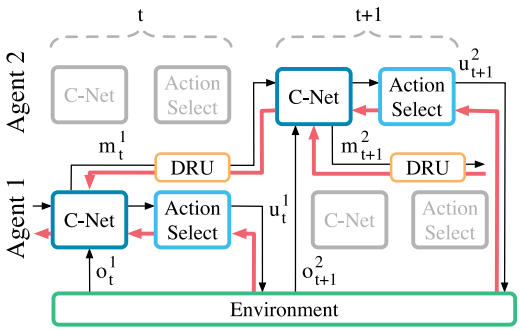

Abbildung 2.2: Die Funktionsweise von DIAL [2, S. 4]

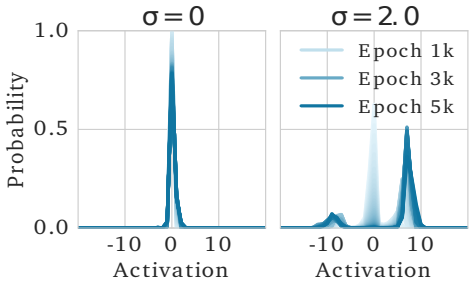

Abbildung 2.3: Auswirkungen des Rauschens der DRU auf der gelernten Aktivierungen der DIAL Agenten [2, S. 8]

Nachricht. Den Gradienten für diese Nachricht kann der Agent der sie abgeschickt hat nun wiederum für seine eigene Backpropagation verwenden. Auf diese Weise entsteht eine Ende-zu-Ende Backpropagation zwischen den Agenten. Die Belohnung, die ein Agent zum Zeitpunkt $t + 1$ für seine gewählte Aktion erhält, wirkt sich also auf die Agenten aus, die ihm zum Zeitpunkt t eine Nachricht gesendet haben. Die Belohnung die diese Agenten auf ihre Aktionen zum Zeitpunkt t erhalten haben, wirken sich zusammen mit dem Gradienten der von ihnen gesendeten Nachricht wiederum auf die Agenten aus, die ihnen eine Nachricht gesendet haben. Dies setzt sich so bis in den Startzustand fort. Somit haben nicht nur vergangene Aktionen Einfluss auf die Zukunft, sondern auch vergangene Nachrichten.[2, S. 5]
Die Anpassungen, die die DRU an den Nachrichten vornimmt, ist für ein erfolgreiches Training äußerst wichtig. Während des Trainings kommunizieren die Agenten mit reellen Zahlen. Bei der späteren Ausführung soll jedoch nur mit Einsen und Nullen kommuniziert werden. Diese Differenzierung der Nachrichten findet in der DRU statt. Aus Sicht der Agenten macht es keinen Unterschied, ob sie im Trainings- oder Inferenz-Modus ausgeführt werden. Damit die Agenten lernen Nachrichten zu versenden, die die DRU korrekt differenzieren kann, regularisiert sie die Nachrichten während des Trainings. Hierfür fügt sie der Nachricht m_t ein Rauschen hinzu. Die Funktion für die Regularisierung lautet $DRU(m_t) = Logistic(N(m_t, \sigma))$ mit σ als Standardabweichung des Rauschens. Die neue Nachricht $DRU(m_t)$ wird daraufhin an die anderen Agenten weitergeleitet.[2, S. 5] Je größer das gewählte σ ist, desto größer ist das Rauschen. Während sich bei einem σ von 0 die Nachrichten sehr ähneln, entstehen bei einem σ von 2 beispielsweise zwei eindeutig unterscheidbare Wertebereiche (siehe Abbildung 2.3).[2, S. 12]
Diese Technik funktioniert dadurch, dass die Agenten die Regularisierung durch ihre gewählten Nachrichten ausgleichen müssen. Geht man z. B. davon aus, dass ein Agent

normalerweise die Nachrichten 1 und 3 senden würde und die Regularisierung diese Nachricht um bis zu ±5 verfälschen kann, könnte die DRU eine gesendete 1 durch Regularisierung in eine 4 umgewandeln. Der Empfänger würde diese Nachricht falsch interpretieren und wie bei einer erhaltenen 3 handeln. Damit dies nicht passieren kann, passt der Agent sein neuronales Netz so an, dass er stattdessen die Nachrichten -10 und 11 versendet. Nun kann die Regularisierung nicht mehr dafür sorgen, dass die Nachricht falsch interpretiert wird. Während der Ausführung diskretisiert die DRU die Nachrichten stattdessen. Die Umwandlung in Einsen und Nullen erfolgt mit folgender Formel $DRU(m_t) = \mathbb{1}\{m_t > 0\}$.[2, S. 5]

3 Versuchsaufbau

Bei allen Versuchen kommunizieren immer drei Agenten miteinander. Auf diese Weise wird die Anzahl an benötigten Trainingsschritten und der damit verbundenen Zeitaufwand für die einzelnen Versuchsdurchläufe möglichst gering gehalten. Die Agenten verwenden zum Kommunizieren DIAL. DIAL wird ohne parameter sharing eingesetzt. Auf diese Weise wird sichergestellt, dass neue Agenten nicht direkt durch das in vorherigen Versuchen erlangte Wissen profitieren, sondern nur indirekt durch das Handeln der anderen Agenten. Die von den Agenten zu lösende Aufgabe ist das Schalterrätsel, welches in Abschnitt 3.1 genauer beschrieben wird. Die Agenten werden mit Signalgrößen von einem Bit und von vier Bit getestet. Die Signalgröße von einem Bit wurde gewählt, da dies die kleinstmögliche Größe für versendete Nachrichten ist. Die vier Bit Signalgröße wurde gewählt, da mit ihr achtmal so viele individuelle Nachrichten versandt werden können, wie mit einer ein Bit Signalgröße. Um den Aufwand der Nachrichtenauswertung handhabbar zu halten, wurde keine größere Signalgröße als vier Bit verwendet. Die Agenten werden auf folgende Weise getestet:

Versuch 1: Die Agenten lernen das Environment mit den beiden Signalgrößen.

Versuch 2: Von den AmV aus dem ersten Versuch, wird ein Agent durch einen neuen Agenten ersetzt.

Versuch 3: Von den AmV aus dem ersten Versuch, werden zwei Agenten durch neue Agenten ersetzt.

Versuch 4: Ein Agent aus dem ersten Versuch wird durch einen Agenten (ebenfalls aus dem ersten Verusch) mit einem anderen Kommunikationsprotokoll ersetzt. Ein weiterer Agent wird durch einen Agenten ohne Vorwissen ersetzt. Bei den AmV wird der Optimizer zurückgesetzt.

Versuch 5: Wiederholung von den Versuchen zwei und drei. Diesmal wird der Optimizer der AmV zurückgesetzt.

3.1 Environment

Bei dem Schalterrätsel handelt es sich um ein Environment, welches auch bei den Versuchen von Foerster et al. zu DIAL verwendet wurde.[2, S. 6] In diesem Rätsel haben Gefangene die Chance auf Freiheit. Wenn alle Gefangenen verhört wurden, müssen sie dies dem Gefängniswärter mitteilen. Auf diese Weise erhalten sie die Freiheit. Dabei können die Gefangenen nicht direkt miteinander reden. Jeden Tag wird ein Gefangener aus seiner Zelle in den Verhörraum geführt. In diesem Verhörraum steht eine Lampe, die er ein- oder ausschalten kann. Wenn nun am folgenden Tag ein Gefangener in den Verhörraum geführt wird, kann er sehen, ob die Lampe leuchtet oder nicht. Auf diese Weise können die Gefangenen miteinander kommunizieren. Welcher Gefangene in den Verhörraum geführt wird, ist immer zufällig. Nur der Gefangene im Verhörraum kann angeben, ob er glaubt, dass bereits alle Gefangenen verhört wurden. Wenn er angibt, dass noch nicht alle Gefangenen verhört wurden, geschieht nichts. Gibt er jedoch an, dass bereits alle Gefangenen verhört wurden und die Angabe ist korrekt, kommen alle Gefangenen frei. Stimmt seine Aussage in diesem Fall nicht, werden alle exekutiert.
Die Lampe im Verhörraum wird durch die Nachrichten 1 und 0 repräsentiert. Bei einer Signalgröße von vier Bit steht nicht nur eine Lampe, sondern vier Lampen im Verhörraum. Die DIAL-Agenten kommunizieren bei diesem Environment immer nur mit dem Agenten, der als nächstes in den Verhörraum geführt wird. Wird ein Agent mehrfach hintereinander in den Verhörraum geführt, kommuniziert er mit sich selbst. Die Aufgabenstellung ist kooperativ. Alle Agenten erhalten die gleichen Belohnung. Sie erhalten eine Belohnung von 1, wenn sie erfolgreich sind und eine Belohnung von -1, wenn sie scheitern. Um den Zeitaufwand des Trainings zu verkürzen, wird das Environment, wie bei Foerster et al., automatisch nach sechs Zeitschritten beendet. Geschieht dies, erhalten die Agenten eine Belohnung von 0.

3.2 Agent

Der Aufbau der Agenten erfolgt nach den Vorgaben von Foerster et al..[2, S. 6] Die Eingabeparameter eines Agenten a bestehen aus der Beobachtung o_t^a, der eingehenden Nachricht $m_t^{a'}$ und der letzten Aktion des Agenten u_{t-1}^a. Im Gegensatz zu Foerster et al. erhält der Agent keine Kennung als Eingabeparameter, da kein parameter sharing betrieben wird. Die Parameter u_{t-1}^a und o_t^a werden durch embedding layer mit einer

Ausgangsdimension von 128 Neuronen geleitet. $m_t^{a'}$ wird durch einschichtiges Multi-Layer-Perzeptron (MLP) verarbeitet, dass ebenfalls eine Ausgangsdimension von 128 Neuronen besitzt. Die dabei entstehenden Werte werden zu einem Vektor z_t^a addiert. Dieser Vektor wird zusammen mit h_{t-1}^a an eine zweischichtige Gated Recurrent Unit (GRU) übergeben, sodass $h_t^a = GRU[128, 128](z_t^a, h_{t-1}^a)$ ist. Die Ausgabe der GRU wird an ein zweischichtiges MLP weitergegeben. Dieses MLP produziert die zwei Ausgaben Q_t^a und m_t^a. Als Optimizer für die Agenten wird der Root Mean Square Propagation (RMSProp) Optimizer verwendet. RMSProp arbeitet mit einem exponetiell geglätteten Durchschnitt, um die Lernrate des neuronalen Netzes anzupassen.[4, S. 15] Das Zurücksetzen des Optimizers bei einem Teil der Versuche sorgt dafür, dass die Agenten ihren Durchschnitt nur mit den neuen Werten berechnen. Die Lernrate für AmV wird bei zurückgesetztem Optimizer also nicht von vorherigen Versuchen beeinflusst. Die Lernrate für AmV ist bei zurückgesetztem Optimizer somit anfangs potentiell größer. Pro Trainingsepisode werden die Agenten mit je 32 Batches trainiert.

4 Ergebnisse

Im Folgenden wird zuerst der Belohnungsverlauf analysiert. In diesem werden die erhaltenen Belohnungen normalisiert. Das bedeutet in diesem Fall, dass Durchläufe, bei denen die Agenten nicht gewinnen können, weil nie alle Agenten den Verhöhrraum betreten, bei der Berechnung des arithmetischen Mittels nicht mit einbezogen werden. Danach werden die Kommunikationsprotokolle der Agenten analysiert. Bei allen Versuchen werden Agenten als AmV angesehen, wenn sie 5000 Trainingsepisoden durchlaufen haben. Nach dieser Zeitspanne ist keine große Fluktuation bei den erhaltenen Belohnung der Agenten zu erkennen. Es wurde für jeden Test der Mittelwert aus zehn Versuchen gebildet. Des Weiteren werden Agenten, mit Ausnahme der AmV, nur für 2000 Episoden trainiert. Diese Anzahl an Trainingsepisoden hat sich als ausreichend herausgestellt, um Aussagen zur Lerngeschwindigkeit zu treffen. In der weiteren Analyse werden Agenten mit einer Signalgröße von einem Bit als „1-Bit Agenten“ bezeichnet, bei einer Signalgröße von vier Bit als „4-Bit Agenten“.

4.1 Analyse der Belohnungsverläufe

In den folgenden Unterabschnitten werden die Belohnungsverläufe aller Versuche analysiert. Dabei wird jeder Versuch direkt mit dem ensprechenden Versuch mit zurückgesetztem Optimizer verglichen. Im letzten Unterabschnitt werden die Ergebnisse der Belohnungsverläufe untereinander verglichen.

4.1.1 Versuch 1 - Agenten ohne Vorwissen

Wie in Abbildung 4.4 zu erkennen ist, ist der Belohnungsverlauf sowohl bei Agenten mit einer Signalgröße von einem Bit, als auch bei denen mit einer Signalgröße von vier Bit, beim ersten Versuch in etwa gleich. Ab ca. 250 Trainingsepisoden ist eine starke Steigerung bei den erhaltenden Belohnungen zu erkennen. Die 1-Bit Agenten erhalten

dabei bis etwa zur 600. Episode eine etwas größere Belohnung als die 4-Bit Agenten. Bis zur 1000. Episode erhalten die Agenten in etwa die gleiche Belohnung. Ab der 1000. Episode erhalten die 4 Bit Agenten eine größere Belohnung. Des Weiteren sinkt die Standardabweichung drastisch und ist bei der 2000. Episode kaum noch vorhanden. Die 1-Bit Agenten haben hingegen bis zur 2000. Episode eine hohe Standardabweichung und erreichen im Mittel keine perfekte Belohnung. Dies steht im Gegensatz zu der in der zweiten Hypothese aufgestellten Vermutung, da die 4-Bit Agenten in diesem Fall schneller perfekte Resultate erzielen, als die 1-Bit Agenten.

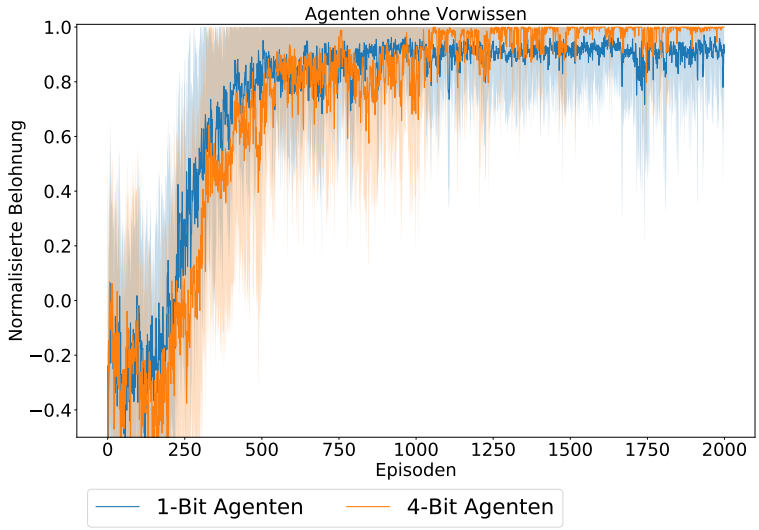

Abbildung 4.1: Performance von DIAL, ohne AmV, mit Standardabweichung.

4.1.2 Versuch 2 - Zwei Agenten mit Vorwissen

Der Belohnungsverlauf der 1- und 4-Bit Agenten unterscheidet sich kaum. Nach ca. 1250 Episoden erreichen die 1-Bit Agenten häufig perfekte Resultate, während die 4-Bit Agenten bis zur 2000. Episode immer noch eine hohe Standardabweichung haben. Wenn der Optimizer des neuralen Netzes der Agenten zurückgesetzt wird, ist jedoch das genaue Gegenteil der Fall. Während die 4-Bit Agenten bereits nach 750 Episoden anfangen, sich auf einer perfekten Belohnung zu stabilisieren, erreichen die 1-Bit Agenten diese Resultate auch nach 2000 Episoden noch nicht. Zwar verkleinert sich die Standardabweichung der 1-Bit Agenten nach ca. 1600 Episoden, sie erreichen jedoch im Durchschnitt keine perfekten Ergebnisse. Die 4-Bit Agenten erhalten hingegen schon nach 1250 Episoden fast ausschließlich perfekte Ergebnisse.

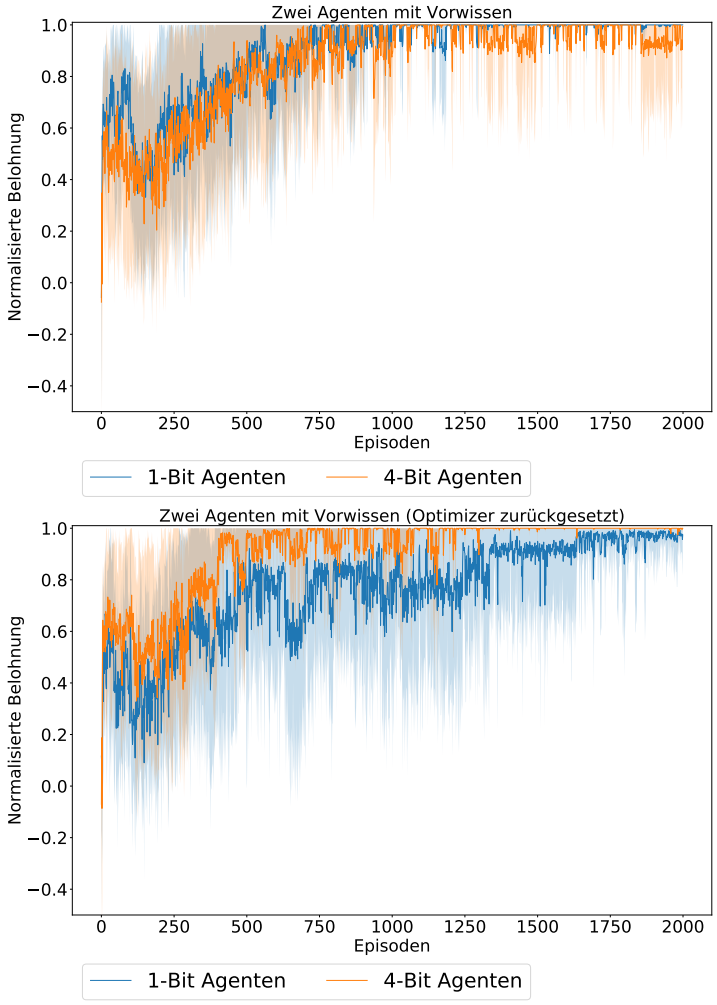

Abbildung 4.2: Performance von DIAL mit zwei AmV, ohne (oben) und mit zurückgesetztem Optimizer (unten), mit Standardabweichung.

4.1.3 Versuch 3 - Ein Agent mit Vorwissen

Der Belohnungsverlauf der 1- und 4-Bit Agenten unterscheidet sich erneut kaum. Beide erhalten nach ca. 600 Episoden häufiger perfekte Belohnungen. Nach ca. 1500 Episoden erhalten beide fast ausschließlich perfekte Ergebnisse, jedoch stabilisieren sich die 4-Bit Agenten etwas schneller auf der perfekten Belohnung. Wenn der Optimizer der Agenten zurückgesetzt wird, hat das in diesem Fall sowohl auf die 1-Bit Agenten, als auch auf die 4-Bit Agenten einen negativen Effekt. Zum einen ist die Ausgangsbelohnung in beiden Fällen geringer, zum anderen ist die durchschnittlich erhaltene Belohnung auch in den späteren Episoden noch starken Schwankungen ausgesetzt. Bei zurückgesetztem

Optimizer erhalten 1-Bit Agenten erst nach ca. 1400 Episoden zum ersten Mal perfekte Belohnungen. Bei 4-Bit Agenten ist dies zwar schon nach ca. 700 Episoden der Fall, allerdings schwanken die erhaltenen Belohnungen auch noch bei 2000 Episoden. Die Performance der 4- Bit Agenten ist mit zurückgesetztem Optimizer besser, als die der 1-Bit Agenten.

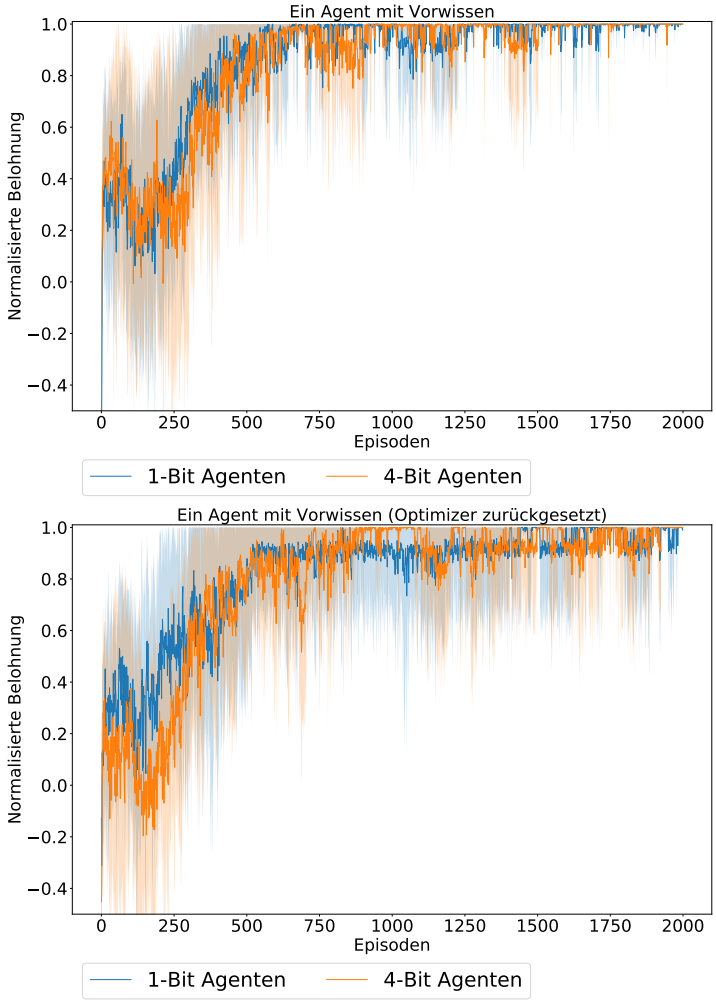

Abbildung 4.3: Performance von DIAL mit einem AmV, ohne (oben) und mit zurückgesetztem Optimizer, (unten) mit Standardabweichung.

4.1.4 Versuch 4 - Verschiedene Kommunikationsprotokolle

Wenn die AmV unterschiedliche Kommunikationsprotokolle gelernt haben, ist die Ausgangsbelohnung der 1-Bit und 4-Bit Agenten höher als bei den Versuchen ohne AmV. Ansonsten schneiden die 1-Bit Agenten sehr schlecht ab. Sie erzielen im Durchschnitt ungefähr die gleichen Ergebnisse wie die Agenten von Foerster et al., die nicht kommunizieren.[2, [S. 7] Auch nach 2000 Episoden ist die Standardabweichung der erhaltenen Belohnungen der 1-Bit Agenten sehr hoch. Die 4-Bit Agenten erhalten hingegen bereits nach ca. 850 Episoden das erste Mal eine durchschnittlich perfekte Belohnung. Nach ca. 1100 Episoden erhalten sie fast ausschließlich perfekte Belohnungen.

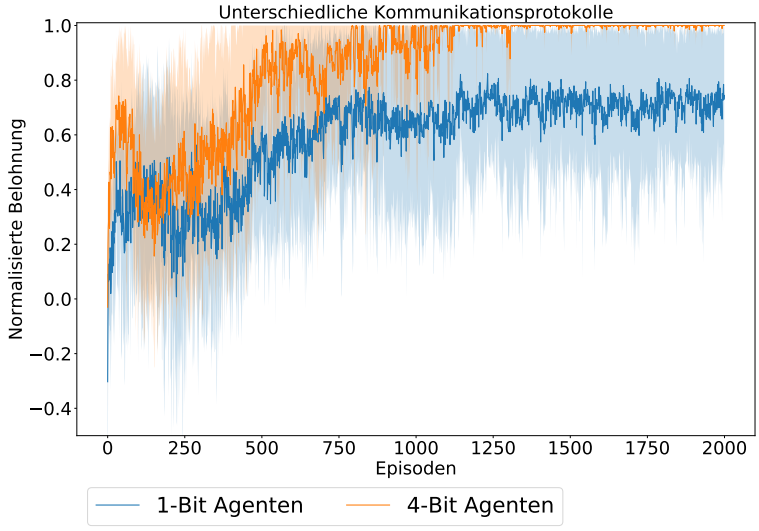

Abbildung 4.4: Performance von DIAL mit zwei AmV, die verschiedene Kommunikationsprotokolle gelernt haben, mit Standardabweichung.

4.1.5 Vergleich der Ergebnisse

Wie zu erwarten, erhalten Agenten bei den Versuchen mit AmV eine deutlich höhere Ausgangsbelohnung (siehe Abbildung 4.5). Bei zwei AmV ist sie höher als bei einem AmV. Diese bessere Leistung liegt wahrscheinlich unter anderem daran, dass die AmV einen Durchlauf nie vor dem dritten Zeitschritt beenden und so die Chance auf ein zufälliges korrektes Beenden größer ist. Dass die 4-Bit Agenten mit zwei AmV bei der Ausgangsbelohnung schlechter abschneiden als die 1-Bit Agenten, könnte daran liegen,

dass es bei einer Signalgröße von einem Bit wahrscheinlicher ist, dass der Agent ohne Vorwissen zufällig die korrekte Nachricht abschickt.
Zusätzlich ist in Abbildung 4.5 ist zu erkennen, dass die 1-Bit Agenten deutlich von AmV zu profitieren scheinen. Während drei Agenten ohne Vorwissen nach 2000 Episoden im Durchschnitt nicht perfekt belohnt werden, ist dies bei Versuchen mit AmV schon nach 700 Episoden das erste Mal der Fall. Dies spricht für die erste Hypothese

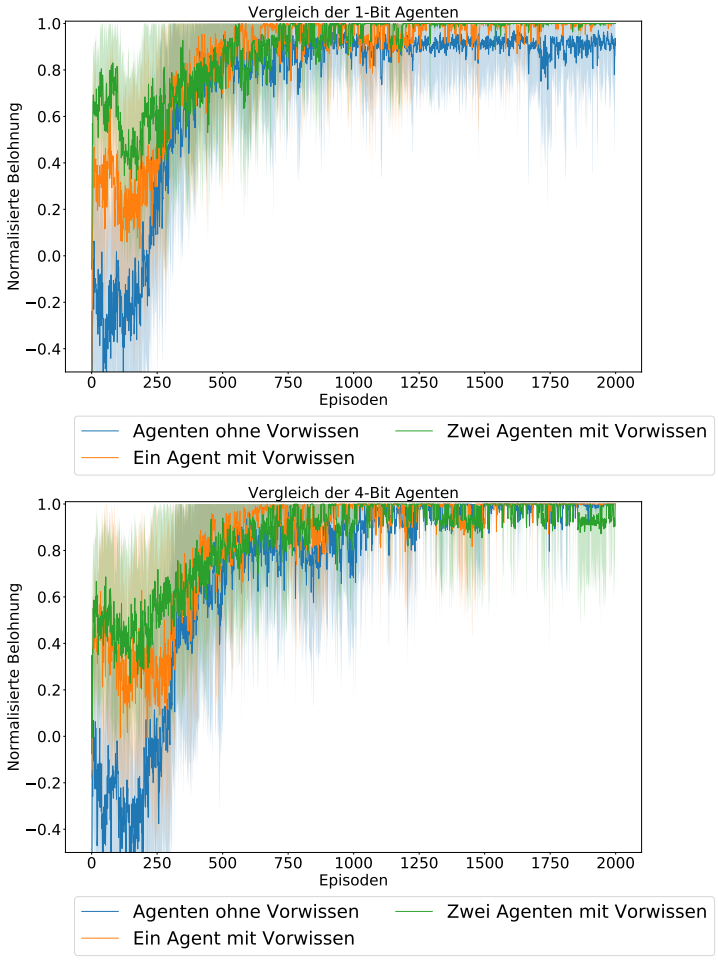

Abbildung 4.5: Performance von DIAL mit keinem, einem und zwei AmV, bei einer Signalgröße von einem Bit (oben) bzw. vier Bit (unten), mit Standardabweichung.

die besagt, dass die Beteiligung von AmV das Lernen beschleunigt. Dieser Vorteil fällt jedoch bedeutend geringer aus, wenn der Optimizer der Agenten zurückgesetzt wird. Im Fall von zwei AmV ist die anfängliche Performance sogar schlechter als bei keinem Agenten mit Vorwissen. Der Belohnungsverlauf aller Versuche mit 1-Bit Agenten ist innerhalb der ersten 200 Episoden ungefähr gleich. Danach steigt die erhaltene Beloh-

nung bei den Versuchen ohne AmV am schnellsten, bei zwei AmV am langsamsten. Dies könnte daran liegen, dass sich in einem Fall drei Agenten stark verbessern und im anderen nur ein Agent. Des Weiteren ist der Spielraum für Verbesserungen deutlich geringer, da die Ausgangsbelohnung bei Beteiligung von AmV deutlich größer ist. Diese Annahmen werden dadurch bestärkt, dass sich die Steigung angleicht, sobald die gleiche durchschnittliche Belohnung erreicht wird. Allerdings stabilisieren sich die Agenten bei den Versuchen mit AmV schneller auf der perfekten Belohnung, selbst wenn ihr Optimizer zurückgesetzt wird.
Bei Agenten mit einer Signalgröße von vier Bit ist ein Vorteil durch AmV nicht so eindeutig zu erkennen. Bei Versuchen mit nur einem AmV werden zwar merklich bessere Ergebnisse erzielt, bei zwei AmV verschlechtern sich die Ergebnisse allerdings. Diese Agenten erreichen dagegen früher perfekte Belohnungen, sind jedoch auch nach 2000 Episoden immer noch instabiler als bei Versuchen ohne AmV. Dieser Unterschied fällt vor allem beim Vergleich mit einem AmV auf. In diesem Fall stabilisieren sich die Agenten schon nach ca. 1500 Episoden. Betrachtet man allerdings die Versuche, bei denen der Optimizer zurückgesetzt wird, sieht man herausragende Ergebnisse bei den Versuchen mit zwei AmV. Diese Agenten liefern die besten Ergebnisse aller Versuche und sind schon nach nur 1250 Episoden in einem optimalen Zustand. Dafür schneiden in diesem Fall die Agenten, bei denen nur ein AmV beteiligt ist, deutlich schlechter ab. Sie sind nach 2000 Episoden instabiler als ihr Pendant ohne zurückgesetzten Optimizer nach 900 Episoden. Die negativen Auswirkungen auf die 4-Bit Agenten sind jedoch kleiner als bei den 1-Bit Agenten, welche sich, bei zurückgesetztem Optimizer, in jedem Fall verschlechtern.
Bei den Versuchen mit unterschiedlichen Kommunikationsprotokollen ist sowohl die Ausgangsbelohnung der 1-Bit Agenten, als auch die der 4-Bit Agenten höher als bei Versuch 1. Sie ähnelt der Ausgangsbelohnung der Versuche mit zwei AmV mit zurückgesetztem Optimizer. Der Belohnungsverlauf der 4-Bit Agenten ähnelt ebenfalls sehr dem Belohnungsverlauf der 4-Bit Agenten aus dem Versuch, bei dem zwei AmV mit zurückgesetztem Optimizer beteiligt sind. Somit lässt sich die vierte Hypothese nicht eindeutig bestätigen. Bei den 1-Bit Agenten zwar eine drastische Verschlechterung zu erkennen, die 4-Bit Agenten verbessern sich jedoch deutlich.
Wenn man bei den Versuchen eins bis drei und fünf die Ergebnisse der 1-Bit Agenten mit denen der 4-Bit Agenten vergleicht (siehe Abbildung 4.6), ist kaum ein Unterschied zwischen den 1-Bit und den 4-Bit Agenten zu erkennen. Dies spricht gegen die zweite Hypothese, dass Agenten mit einer größeren Signalgröße immer eine höhere Anzahl an Trainingsepisoden zum Lösen einer Aufgabe benötigen.

Für 1-Bit Agenten trifft die erste Hypothese, dass Agenten schneller lernen, wenn AmV beteiligt sind, bei allen Versuchen zu. Bei den 4-Bit Agenten lässt sich diese Aussage nicht so leicht treffen. Hier hängt es stark davon ab, wie viele Agenten Vorwissen haben und ob der Optimizer zurückgesetzt wird. Damit wird die Hypothese 1a widerlegt. Bei Agenten mit größeren Signalgrößen ist der verstärkte Lerneffekt nicht stärker ausgeprägt. Er ist in manchen Fällen sogar gar nicht vorhanden.
Dass sowohl die größere Signalgröße, das Zurücksetzen des Optimzers und die Anzahl an AmV manchmal von Vorteil und manchmal von Nachteil sind, deutet darauf hin, dass sie, wie die anderen Parameter eines neuronalen Netzes, der entsprechenden Situation angepasst werden müssen. Aus dem vierten Versuch geht jedoch hervor, dass eine größere Signalgröße von Vorteil ist, wenn Agenten mit verschiedenen Kommunikationsprotokollen miteinander interagieren. Aus dem Versuch ist nicht erkennbar, ob die 4-Bit Agenten genauso schlecht wie die 1-Bit Agenten abschneiden würden, wenn ihr Kommunikationsprotokoll keine Überschneidungen hätte. Jedoch ist bei einer größeren Signalgröße die Wahrscheinlichkeit höher, dass eine solche Überschneidung vorhanden ist.

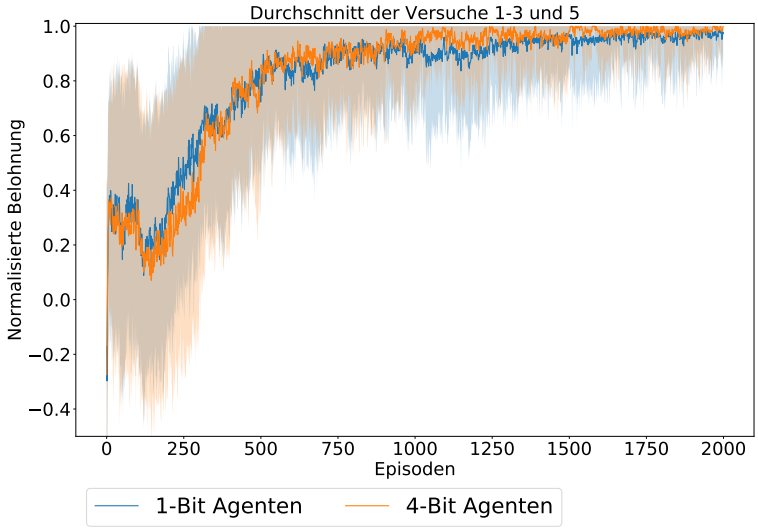

Abbildung 4.6: Durchschnittliche Performance der Versuche 1-3 und 5 mit DIAL, mit Standardabweichung.

4.2 Analyse der Kommunikationsprotokolle

Bei der folgenden Analyse der Kommunikationsprotokolle werden die drei Agenten als Agent A, Agent B und Agent C bezeichnet. In den Fällen mit einem AmV ist dies Agent A. Bei zwei AmV sind es Agent A und Agent B.

4.2.1 1-Bit Agenten

Die Agenten mit einer Signalgröße von einem Bit haben sich, bei den Versuchen ohne Vorwissen, auf zwei unterschiedliche Protokolle geeinigt. Dabei haben sich die Agenten bei acht von zehn Versuchen auf das gleiche Protokoll geeinigt, wie bei Foerster et al.. Dieses lässt sich auf folgende Weise umschreiben:

1. Der erste Agent, der den Verhörraum betritt, sendet eine 1.
2. Wenn ein Agent bereits einmal in einem Verhörraum war, sendet er die Nachricht, die er gerade empfangen hat.
3. Wenn ein Agent zum ersten Mal den Verhörraum betritt und eine 1 empfängt, sendet er eine 0.
4. Wenn ein Agent zum ersten Mal den Verhörraum betritt und eine 0 empfängt, gibt er an, dass alle Agenten bereits im Verhörraum waren.

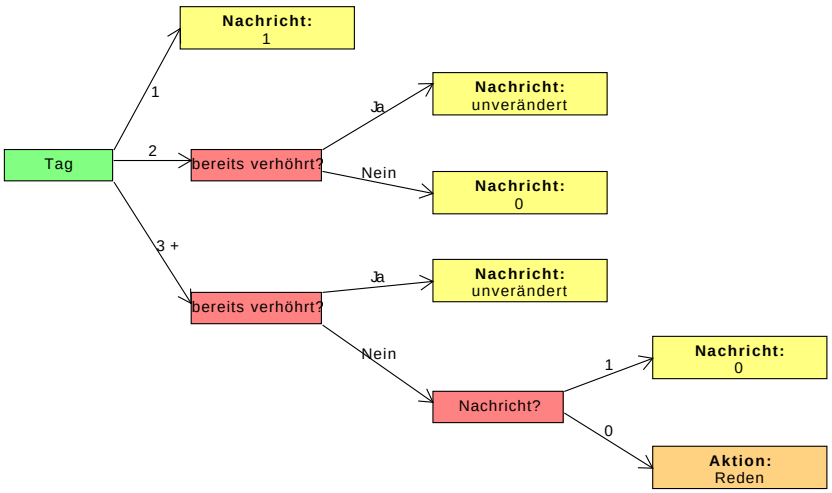

Abbildung 4.7: Kommunikationsprotokoll der 1-Bit Agenten zum Lösen des Schalterrätsels.

Das Protokoll, auf das sich die anderen Agenten einigten, ist im Kern das gleiche Protokoll. Es unterscheidet sich dadurch, dass im ersten Schritt eine 0 statt einer 1 gesendet wird. Bei den nachfolgenden Schritten werden bei den erwarteten und gesendeten Nachrichten ebenfalls 1 und 0 vertauscht. Bei allen Versuchen, bei denen bereits zwei Agenten Vorwissen haben, hat der dritte Agent immer das Kommunikationsprotokoll der anderen übernommen. Bei den Versuchen, bei den nur ein Agent Vorwissen hat, haben die beiden anderen Agenten ebenfalls das Protokoll übernommen. Dies war sogar dann der Fall, wenn bei den Versuchen der Optimizer zurückgesetzt wurde. Dadurch wird die dritte Hypothese für 1-Bit Agenten bestätigt.
Bei Versuch 4, bei dem die AmV unterschiedliche Kommunikationsprotokolle benutzen, einigen sich die Agenten lediglich bei zwei der zehn Versuchsreihen auf ein einheitliches Protokoll. Dabei wurde entweder das Protokoll von Agent A oder Agent B von den anderen beiden Agenten übernommen. Bei den anderen Versuchsreihen übernimmt teilweise Agent A das Protokoll von Agent B, während Agent B zur gleichen Zeit das Protokoll von Agent A übernimmt. Häufig wird auch nur die Hälfte des Protokolls eines anderen Agenten übernommen. Agent C übernimmt in der Hälfte der Versuchsreihen das Protokoll eines anderen Agenten. In den anderen Versuchsreihen werden die Nachrichten ebenfalls komplett übernommen, jedoch wird z. B. die erste Nachricht von Agent A und die zweite von Agent B übernommen. Mit dem dabei entstehenden Mischprotokoll würden die 1-Bit Agenten selbst dann nicht erfolgreich sein, wenn alle Agenten das gleiche Protokoll sprächen. Das Agent C bei der Hälfte der Versuchsreihen das komplette Kommunikationsprotokoll eines Agenten übernommen hat, stärkt die Bestätigung der dritten Hypothese.

4.2.2 4-BitAgenten

Die Agenten mit einer Signalgröße von vier Bit verwenden im Kern die gleiche Strategie wie die 1-Bit Agenten. Die Agenten senden auch Nachrichtenpaare, wie z.B. 1111 und 0000 oder 1001 und 0010. Jedoch ist es möglich, dass jeder Agent sein eigenes Nachrichtenpaar sendet. Die 4-Bit Agenten sind also in der Lage, Nachrichten zu verstehen, die sie selbst nicht versenden. Zum Teil verändert sich die zweite Nachricht der Agenten bei späteren Zeitschritten. Für die erste Nachricht ist dies nicht der Fall. Das in Unterabschnitt 4.2.1 beschriebene Nachrichtenprotokoll muss für die 4-Bit Agenten also in eine allgemeinere Version abgewandelt werden. Diese lässt sich wie folgt umschreiben:

1. Der erste Agent, der den Verhörraum betritt, sendet seine Nachricht für Schritt 1 (N1).
2. Wenn der gleiche Agent direkt wieder den Verhörraum betritt, wiederholt er N1. Dies geschieht, bis ein anderer Agent den Raum betritt.
3. Wenn ein Agent zum ersten Mal den Verhörraum betritt und die N1 eines anderen Agenten empfängt, sendet er seine Nachricht für Schritt 2 (N2).
4. Wenn ein Agent bereits im Verhörraum war und die N2 eines anderen Agenten empfängt, sendet er seine eigene N2.
5. Wenn ein Agent zum ersten Mal den Verhörraum betritt und die N2 eines anderen Agenten empfängt, gibt er an, dass alle Agenten bereits im Verhörraum waren.

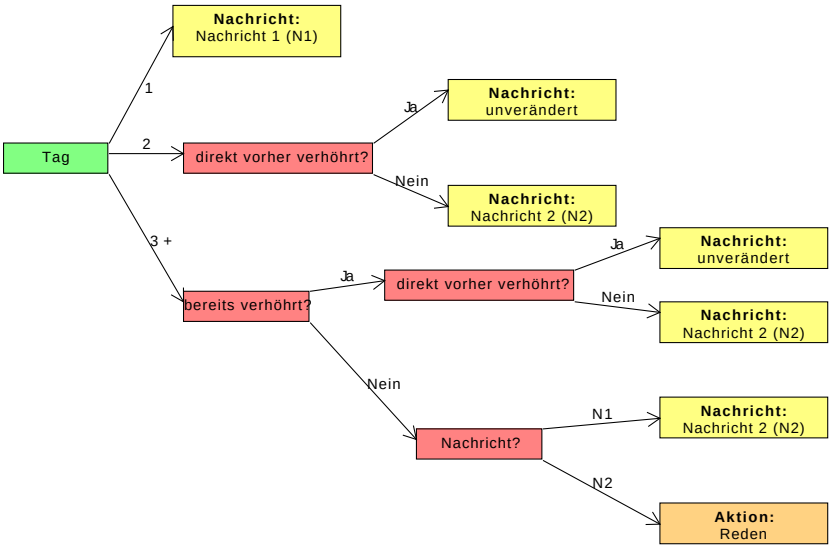

Abbildung 4.8: Kommunikationsprotokoll der 4-Bit Agenten zum Lösen des Schalterrätsels.

Bei genauerer Betrachtung der 4-Bit Protokolle fällt auf, dass für die Entscheidung der Agenten scheinbar nicht immer die ganze Nachricht relevant ist, sondern nur bestimmte Bits. Diese unterscheiden sich bei den Agenten. So erwartet Agent A bspw. die Nachricht 0001, Agent B die Nachricht 1100 und Agent C 1000. Dabei ist es auch möglich, dass ein Agent mit verschiedenen Nachrichten korrekt umgehen kann. So kann Agent C sowohl die Nachricht 1000, als auch die Nachricht 0001 als Zeichen verstehen, dass bereits alle anderen Agenten den Verhörraum betreten haben. All diese Möglichkeiten werden durch die Nachricht 1101 abgedeckt. Die Nachricht 1111 würden die Agenten

in diesem Fall ebenfalls verstehen, solange sie nicht als Vorraussetzung haben, dass das dritte Bit eine Null ist. Dieses Verhalten könnte den 4-Agenten dabei helfen, Agenten mit unterschiedlichen Kommunikationsprotokollen zu handhaben.
Bei der Hälfte der Versuchsreihen ohne Vorwissen versenden alle Agenten pro Versuch die gleichen Nachrichten für alle Schritte. Bei vier Versuchen der anderen Hälfte senden mindestens zwei der Agenten die gleiche zweite Nachricht. Bei der ersten Nachricht ist das nur bei zwei der fünf anderen Versuche der Fall. Bei den Versuchen, bei den alle Agenten die gleiche Nachricht versenden, ist die zweite Nachricht immer das genaue Gegenteil der ersten Nachricht, bspw. 1001 und 0110. Dies ist auch bei der Mehrheit der Agenten der Fall, bei denen jeder unterschiedliche Nachrichten versendet. Bei den Versuchen, bei denen zwei Agenten bereits Vorwissen haben, wurden Agenten aus einer Versuchsreihe gewählt, bei der die Agenten im zweiten Schritt unterschiedliche Nachrichten verschicken, im ersten jedoch die gleiche. In neun der zehn Versuche ist die erste Nachricht aller Agenten gleich. Bei dem Versuch, bei dem dies nicht der Fall ist, sendet nur der Agent C eine andere erste Nachricht ab. Des Weiteren ist es der einzige Versuch dieser Reihe, bei dem der Agent C seine zweite Nachricht von Agent B übernimmt. Bei allen anderen Versuchen der Reihe übernimmt Agent C seine zweite Nachricht von Agent A.
Wenn die Optimizer bei Versuchen mit zwei AmV zurückgesetzt werden, ist die erste Nachricht aller Agenten in allen zehn Versuchen gleich. Zusätzlich übernimmt Agent C die zweite Nachricht erneut von Agent A. Ein Grund hierfür könnte sein, dass der ursprüngliche Agent C aus dem Versuch, dem die AmV entnommen wurden, ebenfalls die gleichen Nachrichten wie Agent A verschickt hat. Es ist also möglich, dass der neue Agent C bevorzugt diese Nachrichten verschickt, weil auch Agent B diese von anderen Agenten erwartet.
Auch bei den Versuchen mit nur einem AmV senden alle Agenten die gleiche erste Nachricht. In acht der zehn Versuche übernehmen sowohl Agent B als auch Agent C die zweite Nachricht von Agent A. In den übrigen Versuchen übernimmt entweder Agent B oder Agent C die Nachricht. In dem Versuch, in dem Agent C die Nachricht übernimmt, übernimmt sie Agent B in einem späteren Zeitschritt. In fast allen Fällen wird also das Kommunikationsprotokoll des AmV übernommen, auch wenn nur ein Agent Vorwissen besitzt. Bei zurückgesetztem Optimizer ist dies ebenfalls in acht der zehn Versuche der Fall. Jedoch kommt es hier häufiger vor, dass Agent A seine zweite Nachricht abändert. Bei der Hälfte der Versuche ist dies der Fall. Die erste Nachricht ist erneut bei allen Agenten die gleiche.
Bei Versuchen, bei denen die AmV unterschiedliche Kommunikationsprotokolle verwen-

den, behalten beide AmV ihr Kommunikationsprotokoll weitestgehend bei. Lediglich Agent A ändert, wie auch in anderen Versuchen, seine zweite Nachricht teilweise ab. Bei fünf der zehn Versuchsreihen übernimmt Agent C das Protokoll von Agent A. Agent C übernimmt nie das Protokoll von Agent B komplett. Stattdessen übernimmt er in drei Versuchsreihen die erste Nachricht von Agent B und die zweite Nachricht von Agent A. In den anderen beiden Versuchsreihen verwendet Agent C zwei neue, eigene Protokolle. Dies bestätigt die Annahme, dass nicht die gesamte Nachricht für die Agenten relevant ist. Bei dem Kommunikationsprotokoll, welches alle Agenten in diesem Fall indirekt verwenden, ist es relevant, dass bei der ersten Nachricht die beiden mittleren Bits 1 sind (z. B. 0110) und bei der zweiten Nachricht 0 (z. B. 1001).
All diese Erkenntnisse bestätigen größtenteils die dritte Hypothese, welche besagt, dass Agenten ohne Vorwissen die Kommunikationsprotokolle der AmV übernehmen. Dass Agent A bei einigen Versuchen als AmV seine Nachricht abändert, könnte daran liegen, dass die zweite Nachricht nach dem ersten Versuch nicht genau festgelegt war. Da die veränderte Nachricht immer die gleiche (1000 statt 0000) ist, könnte das abgeänderte Bit irrelevant für das Protokoll sein.

5 Zusammenfassung

5.1 Fazit

In der vorliegenden Bachelorarbeit wurde das Verhalten von DIAL anhand eines Schalterrätsels analysiert. Dabei wurde insbesondere auf die Auswirkungen von verschiedenen Signalgrößen und AmV eingegangen.

Die Versuche haben gezeigt, dass sich die erste Hypothese (AmV beschleunigen die Lerngeschwindigkeit) zumindest in Teilen bestätigt. Vor allem die 1-Bit Agenten liefern schneller perfekte Ergebnisse, wenn AmV beteiligt sind. Die Vorteile zeigen sich aber vor allem in den späteren Trainingsepisoden. Bei den anfänglichen Episoden hingegen ist, mit Außnahme der höheren Ausgangsbelohnung, kein Vorteil zu erkennen.
Für 4-Bit Agenten lässt sich die Hypothese nicht bestätigen, da sie, je nachdem wie viele AmV beteiligt und ob der Optimizer zurückgesetzt ist, mal besser und mal schlechter abschneiden. Somit ist die Hypothese 1a (bei größeren Signalgrößen ist der Effekt auf Hypothese 1 stärker ausgeprägt) widerlegt.

Auch die zweite Hypothese (größere Signalgrößen erfordern mehr Trainingsepisoden) lässt sich nicht bestätigen. Eine Signalgröße, die größer als zum Lösen einer Aufgabe erforderlich ist, hat in manchen Fällen scheinbar sogar eine positive Auswirkung. So schneiden bspw. bei den Versuchen, bei denen die Agenten kein Vorwissen haben, die 4-Bit Agenten besser ab, als die 1-Bit Agenten.
Es kann also sogar förderlich sein, eine größere Signalgröße als zum Lösen der Aufgabenstellung notwendig, zu wählen. Ob dies nur für das Schalterrätsel Environment der Fall ist oder Allgemeingültigkeit besitzt, müsste durch Versuche mit weiteren Environments getestet werden.

Die dritte Hypothese (Agenten ohne Vorwissen lernen das Kommunikationsprotokoll der AmV) lässt sich hingegen bestätigen. Bei allen Versuchen mit AmV, mit Ausnahme der Versuche mit unterschiedlichen Kommunikationsprotokollen, haben die 1-Bit

Agenten das Protokoll der AmV übernommen.
Auch bei den 4-Bit Agenten ist dies in in 35 der 40 Durchläufe mit AmV der Fall. In den anderen fünf Durchläufen wird das Protokoll entweder in Teilen übernommen oder bei zwei Agenten ohne Vorwissen von einem der Agenten vollständig übernommen.
Auffällig bei den Versuchen mit 4-Bit Agenten ist, dass wenn alle Agenten die gleichen Nachrichten versenden, die zweite Nachricht immer das Gegenteil der ersten Nachricht ist.

Die vierte Hypothese (unterschiedliche Kommunikationsprotokolle verschlechtern Lerngeschwindigkeit) lässt sich lediglich für die 1-Bit Agenten bestätigen.
Bei den 4 Bit Agenten erhöht sich stattdessen die Lerngeschwindigkeit. Ob dies daran liegt, dass die AmV der 4-Bit Agenten kein genau gegenteiliges Kommunikationsprotokoll hatten, müsste durch weitere Versuche überprüft werden.

Die größte Auffälligkeit bei den Trainingsverläufen findet sich bei den Versuchen mit zwei AmV. Während die 4-Bit Agenten hierbei ursprünglich schlechter abschneiden als bei dem Versuch ohne Vorwissen, liefern sie die besten Ergebnisse aller Versuche, wenn ihr Optimizer zurückgesetzt wird.

5.2 Ausblick

Diese Bachelorarbeit überprüft nicht das Verhalten von DIAL in Umgebungen, die eine größere Signalgröße als 1 Bit zur erfolgreichen Kommunikation brauchen. Dies wäre ein interessanter Aspekt für weitere Versuche.
Ein weiterer Aspekt, der in der Arbeit nicht weitergehend untersucht wurde, wäre zu überprüfen, ob die 4-Bit Agenten auch dann bei unterschiedlichen Kommunikationsprotokollen besser abschneiden, wenn die Kommunikationsprotokolle keine Überschneidungen haben.
Des Weiteren wäre es wissenswert DIAL dahingehend zu untersuchen, ob es auch komplexeren Environments funktioniert. Denkbar wäre ein Fangenspiel, bei dem die Fänger kommunizieren, um den Läufer zu erwischen. Auch Environments bei denen die Agenten unterschiedliche Aufgaben haben wären spannend.

Literaturverzeichnis

[1] Foerster, Jakob: *Learning to Communicate with Deep Multi-Agent Reinforcement Learning - Jakob Foerster*. 2020. – URL https://www.youtube.com/watch?v=JUbIWA7gTJ4

[2] Foerster, Jakob N. ; Assael, Yannis M. ; de Freitas, Nando ; Whiteson, Shimon: Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In: *arXiv e-prints* (2016), Mai, S. arXiv:1605.06676

[3] Hausknecht, Matthew ; Stone, Peter: Deep Recurrent Q-Learning for Partially Observable MDPs. In: *arXiv e-prints* (2015), Juli, S. arXiv:1507.06527

[4] Hinton, Geoffrey ; Srivastava, Nitish ; Swersky, Kevin: *Neural Networks for Machine Learning - Lecture 6e - rmsprop: Divide the gradient by a running average of its recent magnitude*. 2012. – URL http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf

[5] Mnih, Volodymyr ; Kavukcuoglu, Koray ; Silver, David ; Graves, Alex ; Antonoglou, Ioannis ; Wierstra, Daan ; Riedmiller, Martin: Playing Atari with Deep Reinforcement Learning. In: *arXiv e-prints* (2013), Dezember, S. arXiv:1312.5602

[6] Mnih, Volodymyr ; Kavukcuoglu, Koray ; Silver, David ; Rusu, Andrei A. ; Veness, Joel ; Bellemare, Marc G. ; Graves, Alex ; Riedmiller, Martin ; Fidjeland, Andreas K. ; Ostrovski, Georg ; Petersen, Stig ; Beattie, Charles ; Sadik, Amir ; Antonoglou, Ioannis ; King, Helen ; Kumaran, Dharshan ; Wierstra, Daan ; Legg, Shane ; Hassabis, Demis: Human-level control through deep reinforcement learning. In: *Nature* 518 (2015), Februar, Nr. 7540, S. 529–533. – URL https://doi.org/10.1038/nature14236

[7] Sutton, Richard S. ; Barto, Andrew G.: *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. – URL http://incompleteideas.net/book/the-book-2nd.html

Glossar

Agenten mit Vorwissen Agenten, die bereits gelernt haben, ein Environment zu lösen.

Agenten ohne Vorwissen Agenten, die noch nichts gelernt haben.

Ausgangsbelohnung Belohnung, die die Agenten in ihren ersten 100 Trainingsepisoden erhalten.

Kommunikationsprotokoll Ablauf der Nachrichten, auf den sich die Agenten geeinigt haben.

Signalgröße Länge einer Nachricht.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Kooperatives Machine Learning mit DIAL: Untersuchungen zur Agentenkommunikation

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original